2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Attitude estimation of a high-yaw-rate Mobile Inverted Pendulum; comparison of Extended Kalman Filtering, Complementary Filtering, and motion capture

Eric Sihite, Thomas Bewley, *UCSD Coordinated Robotics Lab*[1]

***Abstract*— This paper explores the accuracy of several state estimators used in Mobile Inverted Pendulum (MIP) robots. Accurate state estimation is essential for effective feedback stabilization of such vehicles, especially at high spin rates. The MIP estimation techniques compared in this work are the Complementary Filter, the Complementary Kalman Filter, the (proprietary) Digital Motion Processor (DMP) from the (common) TDK InvenSense MPU-9250, and a dynamically modelled Extended Kalman Filter (EKF). We also derive from scratch the equations governing the dynamics of MIPs undergoing high yaw rates, as used by the EKF, using a Lagrangian formulation. The MIP was then controlled at several different yaw rate setpoints, and the tilt angle estimates were compared with the ("ground truth") measurements obtained via motion capture. Our test results indicate that the high yaw rate dynamic EKF and DMP are significantly more accurate than the usual Complementary Filter and planar dynamic EKF. The inaccuracy of the Complementary Filter is likely caused by the IMU not being aligned with the body's center of mass, creating a significant centrifugal force while spinning quickly.**

I. INTRODUCTION

A Mobile Inverted Pendulum (MIP) robot is a feedback-stabilized inverted pendulum that is rigidly mounted to two individually-controllable coaxial wheels. Many groups have designed and stabilized MIP robots, using techniques ranging from PID to State-Space Control. In general, performance of the feedback stabilization algorithm implemented is limited by the estimator, which is very challenging in a small embedded systems with low-cost processors and sensors. In addition, small light robots can accelerate quickly and operate on fast timescales, further complicating their estimation.

In order to overcome these challenges, groups have employed a variety of estimation approaches, such as complementary filters and Kalman Filters [1][2]. However, many of these solutions obtain body angle estimates by treating the accelerometer and gyroscope as a simple inclinometer [1][2]. While a few references have considered body dynamics during the state estimation process, they only used the gyroscope and encoders as measurements. Further, all prior models for estimation that we could find in the literature were based on planar MIP dynamics, under the assumption that any yaw motions of the MIP were decoupled from the longitudinal dynamics. High-performance MIP (and, ultimately, ball-balancing) robots [3][4], undergoing high yaw rates, are not accurately modeled under this assumption. Smooth stabilization of a very aggressive maneuvers of small MIP like this, or a small ball-balancing robot with complex dynamics, will likely be dependant on an accurate situational awareness provided by such state estimators. This motivates the present investigation, which aims to develop improved state estimates by reconciling with the raw sensor measurements with the dynamic equation of motion of the vehicle itself.

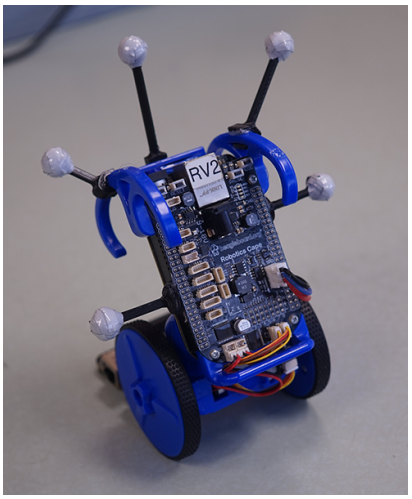

Fig. 1: The eduMIP educational robotics kit developed at the UCSD Coordinated Robotics Lab, available at Renaissance Robotics, with markers for motion capture attached.

In this paper, we present a 3D model for a MIP undergoing fast yaw dynamics, and introduce an Extended Kalman Filter (EKF) for state estimation using both the accelerometer and gyroscope measurements. In addition, using the (commercially-available) Renaissance Robotics eduMIP depicted in Fig. 1 as a test platform and a motion capture system to capture "ground truth", we compared our new yaw model and estimator to three existing methods: the complementary filter, the complementary Kalman filter, and the planar MIP dynamic Kalman Filter.

The most difficult and important state to estimate on a MIP is the body tilt angle θ. The following subsections describe and derive the existing state estimators for θ.

A. Complementary Filter

Complementary Filtering is the process of combining multiple sensor estimate in order to calculate an estimate with less uncertainty than what each individual estimate would

[1]emails: {esihite,bewley}@ucsd.edu

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on December 15,2021 at 03:37:23 UTC from IEEE Xplore. Restrictions apply.

have [5]. The angle θ is estimated using the IMU's gyroscope and acelerometer measurements. The accelerometer gives a noisy θ estimate by obtaining the angle of the gravity vector. The gyroscope outputs a angular velocity ω measurement which can be integrated to obtain a low noise θ estimate that suffers from accumulated integration error. To obtain a better θ estimate, the gyro θ estimate is high-pass filtered added to a low-passed accelerometer estimate [5]. The resulting equation in discrete time which is shown in Eq. 1 is one of the simplest form of estimation for θ.

$$\hat{\theta}_k = c\,\hat{\theta}_{k-1} + \omega_{k-1}^{gyro}\,dt + (1-c)\,\theta_{k-1}^{accel} \tag{1}$$

where $\hat{\theta}$ is the θ estimate, $c = e^{-\sigma dt}$, σ is the high-pass/low-pass cutoff frequency and dt is the controller loop time step.

B. Complementary Filter with Kalman Filter

Complementary Filtering can also be done using Kalman Filter [5]. This filter only uses the measurement from the IMU into the Kalman Filter in order to estimate θ [1][2]. This Complementary Kalman Filter setup is shown in Eq. 2, where $y_k^g = \omega_k + b_k$ is the gyro measurement, ω is the body's true angular velocity and b is the sensor bias. $\theta_k = \theta_{k-1} + \omega_k\,dt$ and y_k^a is the inclinometer angle estimate from the accelerometer measurements, $\boldsymbol{v}_k$ and w_k are zero mean white noise.

$$\begin{aligned} \begin{bmatrix}\theta_k \\ b_k\end{bmatrix} &= \begin{bmatrix}1 & -dt \\ 0 & 1\end{bmatrix}\begin{bmatrix}\theta_{k-1} \\ b_{k-1}\end{bmatrix} + \begin{bmatrix}dt \\ 0\end{bmatrix} y_{k-1}^g + \boldsymbol{v}_k \\ y_k^a &= \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\theta_k \\ b_k\end{bmatrix} + w_k \end{aligned} \tag{2}$$

There are several similar estimation methods that do not use a dynamic model. For example, the state can be estimated using an Indirect Kalman Filter with a kinematic model, which was used on the high performance ball-balancing robot Rezero[4]. There are also other IMU sensor fusion algorithm that are used to estimate the orientation, such as for UAV and wearable sensors [6][7][8]. In addition, the TDK InvenSense MPU-9250, which is the IMU used in our MIP, has a proprietary Digital Motion Processing (DMP) algorithm which gives a very accurate angle estimate without a dynamic model.

C. Dynamic Kalman Filter

The Extended Kalman Filter using the system dynamic model is shown in Eq. 3 with the $\boldsymbol{f}_k$ and $\boldsymbol{h}_k$ are the dynamically modelled system and measurement of the robot. The accuracy of the Kalman Filter's estimate relies heavily on the model.

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{v}_k \\ \boldsymbol{y}_k &= \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k \end{aligned} \tag{3}$$

State estimation using Kalman Filter with the MIP dynamic equation has been done, but mostly in simulations where it is assumed that the θ can be measured directly [9]. In addition, Shimizu et al. developed a MIP Dynamic Kalman Filter using only the gyroscope and encoder measurements [10].

TABLE I: Parameter and Time Varying Variable List. Abbreviations: CoM = Center of Mass, CoR = Center of Rotation (origin of the body frame).

Parameter List	
m_b = body mass.	$\hat{I}_b$ = body inertia about CoM.
m_w = wheel mass.	$\hat{I}_w$ = wheel inertia about CoM.
r = wheel radius.	l = length of body's CoM from CoR.
g = gravity constant.	d = distance between wheels.
k_1 = motor torque gain.	k_2 = motor back EMF gain.

Time Varying Variable List, $i = \{1, 2\}$	
θ = pitch angle.	ϕ_i = wheel i rotation angle.
ψ = yaw angle.	φ_i = wheel i encoder rotation angle.
τ_i = wheel i torque.	u_i = motor i PWM command $\in [-1, 1]$.
$\boldsymbol{p}$ = body CoM position vector.	
$\boldsymbol{p}_{wi}$ = wheel CoM position vector.	
$\boldsymbol{p}_w$ = body frame's origin (CoR) position vector.	
$\boldsymbol{l}_b$ = length vector from CoR to body CoM.	
$\boldsymbol{l}_r$ = length vector of wheel's CoM to the ground.	

The Extended Kalman Filter which uses a 3D high yaw rate MIP dynamic model has not been explored yet. We hypothesized that by implementing this model with the Extended Kalman Filter can yield a significantly more accurate estimates compared to the Complementary filter and the Kalman Filter with planar MIP dynamics. The next section describes the dynamic modeling for MIP robots with high yaw rate.

II. MIP DYNAMIC MODELING

The linearized planar MIP dynamic model in Eq. 4 as derived in [11] is well known and used in MIP robots where the yaw rate is trivial or ignored. Table I lists the parameters used in Eq. 4 and this section.

$$\begin{aligned} \left(\hat{I}_{b1} + m_b L^2\right)\,\ddot{\theta} + m_b r l\,\ddot{\phi} - g\,m_b\,\theta &= -\tau \\ m_b r l\,\ddot{\theta} + \left(\hat{I}_{w1} + (m_w + m_b)\,r^2\right)\,\ddot{\phi} &= \tau \end{aligned} \tag{4}$$

We reformulate the MIP dynamic equation using Lagrangian Dynamics to allow nontrivial yaw rate and determine the body's linear acceleration in order to use the IMU's accelerometer measurements in the Kalman Filter.

A. Frame of References

The MIP kinematics are defined using two separate coordinate frames: the inertial frame and the body frame about the center of rotation as shown in Fig. 2. The body reference frame's axis $\{\boldsymbol{e}^1, \boldsymbol{e}^2, \boldsymbol{e}^3\}$ is pointed to the body's left, back, and top respectively and the variables defined in the body frame are represented with a superscript B, e.g. $\boldsymbol{x}^B$. The rotation from the body into the inertial frame is defined in Eq. 5 below.

$$\begin{aligned} \boldsymbol{x} &= R_B\,\boldsymbol{x}^B \\ R_B &= R_z(\psi)\,R_x(\theta) \end{aligned} \tag{5}$$

where the R_x and R_z are the Euler rotation about the inertial frame's x-axis and z-axis respectively. The linear position

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on December 15,2021 at 03:37:23 UTC from IEEE Xplore. Restrictions apply.

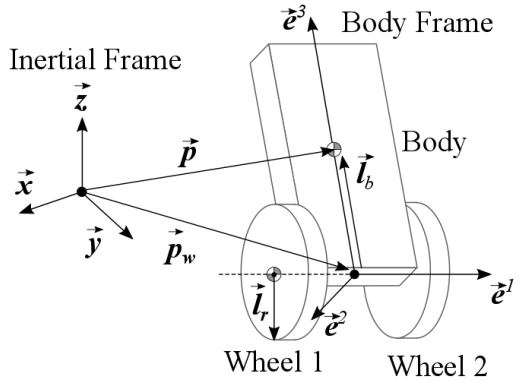

Fig. 2: Coordinate frames of the kinematic model. The body frame origin is the center of rotation for both the body and the wheels.

of the body frame's origin about the inertial frame can be defined with the vector $\boldsymbol{p}_w$ which is the average position of both wheels $\boldsymbol{p}_{w1}$ and $\boldsymbol{p}_{w2}$.

B. Kinematic Formulation

The matrices $\hat{I}_b^B$ and $\hat{I}_w^B$ in Eq. 6 are the moment of inertia about the center of mass in the body frame's orientation for the body and the wheel respectively.

$$\hat{I}_b^B = \begin{bmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{bmatrix} \quad \hat{I}_w^B = \begin{bmatrix} I_{w1} & 0 & 0 \\ 0 & I_{w2} & 0 \\ 0 & 0 & I_{w3} \end{bmatrix} \tag{6}$$

Some of the position and length vectors are defined below:

$$\boldsymbol{p}_w = (\boldsymbol{p}_{w1} + \boldsymbol{p}_{w2})/2 \tag{7}$$

$$\boldsymbol{l}_b^B = \begin{bmatrix} 0 & 0 & l \end{bmatrix}^T, \qquad \boldsymbol{l}_r = \begin{bmatrix} 0 & 0 & -r \end{bmatrix}^T \tag{8}$$

The upper body rotational speed $\boldsymbol{\Omega}$:

$$\boldsymbol{\Omega} = R_z(\psi) \begin{bmatrix} \dot{\theta} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{9}$$

The wheel i rotational speed $\boldsymbol{\omega}_i$:

$$\boldsymbol{\omega}_i = R_z(\psi) \begin{bmatrix} \dot{\varphi}_i \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{\Omega}, \quad i = \{1, 2\} \tag{10}$$

The no slip conditions between the wheel and the ground:

$$d\boldsymbol{p}_{wi}/dt = \boldsymbol{r} \times \boldsymbol{\omega_i}, \quad i = \{1, 2\} \tag{11}$$

$$\psi = (\varphi_1 - \varphi_2)\, r/d \tag{12}$$

Finally, we have the body linear velocity:

$$\frac{d\boldsymbol{p}}{dt} = \frac{d}{dt}\left(R_B \boldsymbol{l}_b^B\right) + \frac{1}{2}\left(\frac{d\boldsymbol{p}_{w1}}{dt} + \frac{d\boldsymbol{p}_{w2}}{dt}\right) \tag{13}$$

C. Lagrangian Dynamics

The MIP equations of motion are derived using Lagrangian dynamics and transformed into the state space form to be used with the Kalman Filter. The kinetic and potential energy into the system are:

$$K_b = \frac{1}{2}(R_B^T\, \boldsymbol{\Omega})^T\, \hat{I}_b\, (R_B^T\, \boldsymbol{\Omega}) + \frac{m_b}{2} \frac{d\boldsymbol{p}}{dt}^T \frac{d\boldsymbol{p}}{dt} \tag{14}$$

$$K_{wi} = \frac{1}{2}(R_B^T\, \boldsymbol{\omega}_i)^T\, \hat{I}_w\, (R_B^T\, \boldsymbol{\omega}_i) + \frac{m_w}{2} \frac{d\boldsymbol{p}_{wi}}{dt}^T \frac{d\boldsymbol{p}_{wi}}{dt} \tag{15}$$

$$U_b = -m_b \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \cdot \boldsymbol{p}, \qquad U_{wi} = 0 \tag{16}$$

$i = \{1, 2\}$. Then the Lagrangian of the system is $L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = K_b + K_{w1} + K_{w2} - U_b$, which is a function of the time varying variable $\boldsymbol{q}$:

$$\boldsymbol{q}(t) = \begin{bmatrix} \theta(t) & \varphi_1(t) & \varphi_2(t) \end{bmatrix}^T \tag{17}$$

We solve for the system dynamic equations using Lagrange's Equation:

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \frac{d}{dt}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{\partial L}{\partial \boldsymbol{q}} - \boldsymbol{\tau} = \boldsymbol{0} \tag{18}$$

Since we applied the constraints into the kinematic equation from Eq. 11 and 12, there is no Lagrange Multiplier in Eq. 18. The force acting onto the system is applied through the wheel's motor. The force $\boldsymbol{\tau}$ and the motor model τ_i are:

$$\boldsymbol{\tau} = \begin{bmatrix} 0 & \tau_1 & \tau_2 \end{bmatrix}^T \tag{19}$$

$$\tau_i = k_1 u_i - k_2 \dot{\varphi}_i, \quad i = \{1, 2\} \tag{20}$$

The wheel angles φ_1 and φ_2 are not useful for controlling the robot. To simplify the states, we modify Eq. 18:

$$\begin{bmatrix} L_1^* \\ L_2^* \\ L_3^* \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 + L_3 \\ L_2 - L_3 \end{bmatrix} = \boldsymbol{0} \tag{21}$$

Next, we do the following change of variables:

$$u_1 = u_x + u_z \qquad \varphi_1 = \varphi + \psi d/(2r) \tag{22}$$

$$u_2 = u_x - u_z \qquad \varphi_2 = \varphi - \psi d/(2r) \tag{23}$$

The equations of motion in Eq. 21 are now a function of $\boldsymbol{q}^* = [\theta, \varphi, \psi]$, $\dot{\boldsymbol{q}}^*$ and $[u_x, u_z]$ where φ is the average encoder angles from both wheels, ψ is the yaw rate, u_x and u_z are the motor command forward and spin respectively. We further simply the equations by assuming the body angle θ and its derivatives are small, similar to our linearized planar MIP equations in Eq. 4. This approximates $\sin\theta \approx \theta$, $\cos\theta \approx 1$, $\{\theta^2, \theta\dot{\theta}, \dot{\theta}^2\} \approx 0$. We can then rearrange Eq. 21 into the general form:

$$M(\boldsymbol{q}^*)\, \ddot{\boldsymbol{q}}^* + \boldsymbol{h}(\boldsymbol{q}^*, \dot{\boldsymbol{q}}^*) = \boldsymbol{\tau}^*(u_x, u_z) \tag{24}$$

$$M\, \ddot{\boldsymbol{q}}^* = \boldsymbol{b} \tag{25}$$

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on December 15,2021 at 03:37:23 UTC from IEEE Xplore. Restrictions apply.

$$M = \begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{26}$$

$$\begin{aligned} m_{11} &= I_{b1} + 2I_{w1} + 2m_w r^2 + m_b(l+r)^2 \\ m_{12} &= m_{21} = 2I_{w1} + 2m_w r^2 + m_b r(l+r) \\ m_{22} &= 2I_{w1} + (m_b + 2m_w)r^2 \\ m_{33} &= I_{w1} d/r + r(dm_w + 2(I_{t3} + 2I_{w3})/d) \\ b_1 &= glm_b\theta + (I_{b2} - I_{b3} + 2(I_{w2} - I_{w3}) + l^2 m_b)\theta\dot{\psi}^2 \\ b_2 &= 2(k_1 u_x - k_2\dot{\varphi}) \\ b_3 &= 2k_1 u_z - k_2 d\dot{\psi}/r \end{aligned} \tag{27}$$

Then we can calculate for our dynamic equation of motion:

$$\boldsymbol{x} = \begin{bmatrix} \theta & \varphi & \psi & \dot{\theta} & \dot{\varphi} & \dot{\psi} \end{bmatrix}^T \tag{28}$$

$$\boldsymbol{u} = \begin{bmatrix} u_x & u_z \end{bmatrix}^T \tag{29}$$

$$\frac{d\boldsymbol{x}}{dt} = \begin{bmatrix} \dot{\boldsymbol{q}}^* \\ \ddot{\boldsymbol{q}}^* \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}}^* \\ M^{-1}\,\boldsymbol{b} \end{bmatrix} = \boldsymbol{f}_c(\boldsymbol{x}, \boldsymbol{u}) \tag{30}$$

The equation above can be simplified down into the linear planar MIP dynamic in Eq. 4 by setting $\dot{\psi} = 0$ and doing a change of variable $\phi = \varphi + \theta$.

We can now solve for the body's linear acceleration as a function of $\boldsymbol{x}$ using the angular acceleration $\ddot{\boldsymbol{q}}^*$ from above. The linear acceleration as measured by the accelerometer is:

$$\boldsymbol{y}_a^B(\boldsymbol{x}) = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = R_B^T \left(\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} - \frac{d^2\boldsymbol{p}}{dt^2} \right) \tag{31}$$

We simplify the $\boldsymbol{y}_a^B$ using the same small θ and $\dot{\theta}$ assumption as the equation of motion above. Only a_y (body forward/back direction) is used in the EKF because the other acceleration measurements were very noisy even when idling. a_y is expressed below:

$$a_y = -l\theta\dot{\psi}^2 + (l+r)\ddot{\theta} + r\ddot{\varphi} - g\theta \tag{32}$$

III. EXPERIMENTAL SETUP

We used motion capture to obtain a ground truth model to test the validity of our various state estimators. We were able to simultaneously we record all the θ estimates on board the robot and the camera data from motion capture. To test the performance of our estimators, we ran our MIP test under three different experiments: idling balancing, 5 rad/s and 10 rad/s spinning while balancing in place.

A. MIP Hardware

The MIP used in this work is the eduMIP, available from Renaissance Robotics, as shown in Fig. 1. The Edu MIP is controlled using the Beaglebone Black and a robotics cape, which contains the 9-axis IMU, breakout for encoder counting and motor drivers. Beaglebone can store data into its hard drive at a sampling rate of 50 Hz. The parameter values for this Edu MIP can be seen in Table II, which were estimated using system identification by Zhuo [12].

TABLE II: Edu MIP Parameter Values.

Parameter	Value	Parameter	Value
m_b	218 g	I_{b1}	5.91 10^{-4} kg.m^2
m_w	24 g	I_{b2}	2.99 10^{-4} kg.m^2
r	35 mm	I_{b3}	2.91 10^{-4} kg.m^2
d	70 mm	I_{w1}	6.42 10^{-5} kg.m^2
l	46 mm	I_{w2}	7.48 10^{-6} kg.m^2
k_1	0.12 N.m	I_{w3}	7.48 10^{-6} kg.m^2
k_2	0.003 N.m.s/rad	g	9.8 m/s^2
dt	0.005 s		

B. Motion Capture

The motion capture system comprised of four Optitrack 13 cameras placed in four different corners at varying heights around the testing platform. The camera have a sampling rate of 120 Hz. During testing, we simultaneously recorded the motion capture data on a laptop and state estimates on board MIP's Beaglebone Black. The data was synchronized in post by matching the measurements when the robot begins balancing from its resting position.

C. Controller

In order to balance the MIP, we used a Successive Loop Closure (shown in Fig. 3) for controlling θ and a simple PD controller for controlling ψ. The controller's discrete time transfer functions are:

$$D_1(z) = \frac{-4.95\,z^2 + 8.86\,z - 3.97}{1.000\,z^2 - 1.48\,z + 0.481} \tag{33}$$

$$D_2(z)P = \frac{-0.189\,z^2 + 0.372\,z - 0.184}{1.000\,z^2 - 1.86\,z + 0.86} \tag{34}$$

$$D_z = 1.0(\psi_r - \psi) + 0.05(\dot{\psi}_r - \dot{\psi}) \tag{35}$$

where D_z is the yaw controller which outputs u_z, ψ_r and $\dot{\psi}_r$ are the reference yaw values. This controller uses the DMP measurement from the IMU and the raw encoder values as the state into the controller. Since we want to compare the accuracy of the estimators relative to each other, we want to use the same controller for all of the test and this controller worked very well.

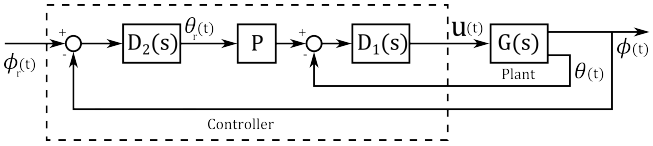

Fig. 3: Successive Loop Closure Block Diagram

D. Complementary and Kalman Filter Setup

This section lists the actual equations used by the estimations. The EKF algorithm follows the standard discrete time EKF algorithm. The controller loop was run at 200 Hz ($dt = 0.005$ s).

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on December 15,2021 at 03:37:23 UTC from IEEE Xplore. Restrictions apply.

1) Complementary Filter: The Complementary Filter was setup with the cutoff frequency of 4 rad/s, which gave us $c = 0.98$ and the following equations:

$$\theta_k^{accel} = \text{atan2}(-a_{yk}, -a_{zk}) \tag{36}$$

$$\hat{\theta}_k = 0.98\,\hat{\theta}_{k-1} + 0.005\,\omega_{k-1}^{gyro} + 0.02\,\theta_{k-1}^{accel} \tag{37}$$

2) Complementary Filter with Kalman Filter: The Complementary Kalman Filter is setup the same way as the Eq. 2, using the following values:

$$y_k^a = \theta_k^{accel} = \text{atan2}(-a_{yk}, -a_{zk}) \tag{38}$$

$$Q_c = \text{diag}([10^{-8}, 10^{-10}]), \qquad R_c = 10^{-4} \tag{39}$$

where the Q_c and R_c are the process and measurement noise covariance matrices respectively.

3) Fast Yaw Dynamic EKF: The equation of motion derived in Eq. 30 is the continuous time dynamic equation, so we need to transform it into a discrete time equation before we can use it in the EKF. The discrete time equation of motion was estimated using the Explicit Euler method. Then the difference equation for the EKF becomes:

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{v}_k \tag{40}$$

$$\boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) = \boldsymbol{x}_{k-1} + dt\boldsymbol{f}_c(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) \tag{41}$$

$$\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \end{bmatrix}^T \tag{42}$$

Plugging in the parameters values in Table II into the equations above yields:

$$f_1 = \theta_k + 0.005\,\dot{\theta}_k \tag{43}$$

$$f_2 = \varphi_k + 0.005\,\dot{\varphi}_k \tag{44}$$

$$f_3 = \psi_k + 0.005\,\dot{\psi}_k \tag{45}$$

$$f_4 = \dot{\theta}_k + 0.629\,\theta_k + 0.068\,\dot{\varphi}_k + 0.009\,\theta_k\,\dot{\psi}_k^2 - 2.72 u_{xk} \tag{46}$$

$$f_5 = 0.813\,\dot{\varphi}_k - 1.12\,\theta_k - 0.016\,\theta_k\,\dot{\psi}_k^2 + 7.47\,u_{xk} \tag{47}$$

$$f_6 = 0.939\,\dot{\psi}_k + 2.44\,u_{zk} \tag{48}$$

and the measurements used in the yaw dynamic EKF is:

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k \tag{49}$$

$$\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \begin{bmatrix} \varphi_k & \omega_k^{gyro} & a_{yk} & \psi_k \end{bmatrix}^T \tag{50}$$

$$\omega_k^{gyro} = \dot{\theta}_k \tag{51}$$

$$a_y = 8.17\,u_x - 7.41\,\theta - 0.204\,\dot{\varphi} - 0.0346\,\theta\,\dot{\psi}^2 \tag{52}$$

The process and measurement noise covariance matrices values used for the EKF are:

$$Q_y = \text{diag}([10^{-11}, 10^{-7}, 10^{-7}, 10^{-5}, 10^{-7}, 10^{-7}]) \tag{53}$$

$$R_y = \text{diag}([10^{-5}, 2.62\,10^{-7}, 5.44\,10^{-3}, 10^{-5}]) \tag{54}$$

4) Planar Dynamic KF: The dynamic equation for the planar MIP dynamic can be calculated from the fast yaw dynamic equation in Eq. 42 by setting $\dot{\psi} = 0$. The dynamic equation will become linear time-invariant and is defined with only 4 states: $\boldsymbol{x} = [\theta, \varphi, \dot{\theta}, \dot{\varphi}]$ and 1 input $u = u_x$. The measurements used in the planar KF is:

$$\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \begin{bmatrix} \varphi_k & \omega_k^{gyro} & a_{yk} \end{bmatrix}^T \tag{55}$$

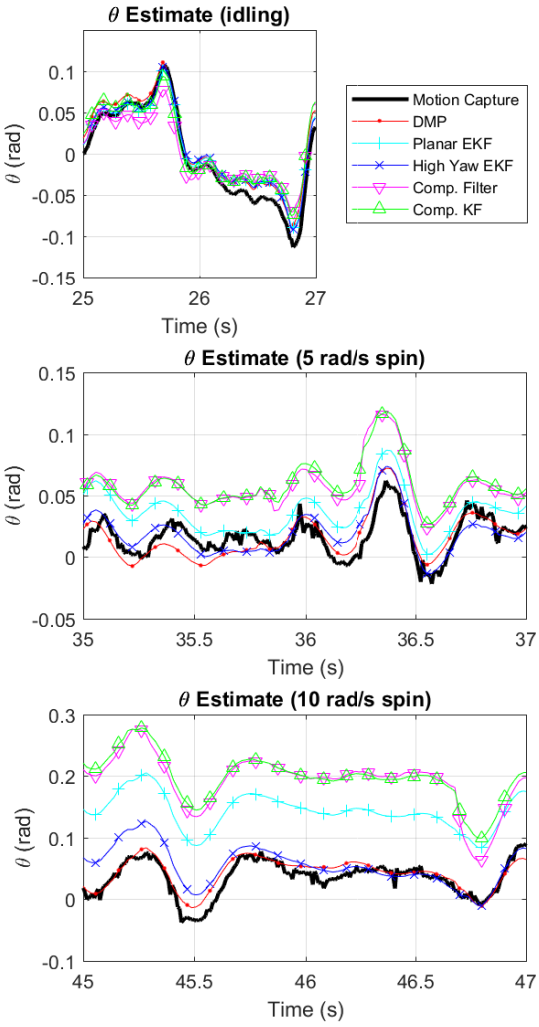

Fig. 4: Estimated θ and the motion captured θ.

TABLE III: Root Mean Squared (RMS) Error between the estimates and the motion captured θ.

Estimator	RMS Error (rad)		
	No Spin	5 rad/s spin	10 rad/s spin
DMP	**0.0148**	**0.0115**	**0.0129**
Planar EKF	0.0133	0.0256	0.1028
High Yaw Rate EKF	**0.0133**	**0.0107**	**0.0249**
Complementary Filter	0.0228	0.0424	0.1605
Complementary KF	0.0199	0.0432	0.1636

The process and measurement noise covariance matrices values are:

$$Q_p = \text{diag}([10^{-11}, 10^{-7}, 10^{-5}, 10^{-7}]) \tag{56}$$

$$R_p = \text{diag}([10^{-5}, 3.07\,10^{-6}, 3.04\,10^{-3}]) \tag{57}$$

IV. RESULT AND DISCUSSION

The θ estimates under several yaw rates can be seen in Figure 4 and their respective Root Mean Squared errors is listed in Table III. The error from Table III shows that the DMP and yaw dynamic EKF have the overall best performance with better accuracy under high yaw rates

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on December 15,2021 at 03:37:23 UTC from IEEE Xplore. Restrictions apply.

compared to the other estimates, like what we expected. The DMP's estimate is especially good because it does not use the system's dynamic equation to get such an accurate estimate. So, it is very likely that we can use the DMP's estimate without the need of using an observer if the orientation is the only important state to estimate.

Looking at the acceleration data in Figure 5 shows an interesting situation. The yaw dynamic EKF's estimated y_3 value did not match the measured value during the 10 rad/s spinning but the estimated θ is still more accurate than the others. We believe that there is an offset on the acceleration data from the IMU placement being not on the top body's center of mass, causing the acceleration data to receive a bias as a function of yaw rate due to the centripetal force. The complementary filter used the atan2 function to determine θ, but the estimation during high yaw rate is heavily skewed due to the accelerator bias and this caused the estimates to be inaccurate during fast yaw movements.

V. CONCLUSION AND FUTURE WORK

In this paper we surveyed several different state estimators used on MIP robots and compared their performance using a ground truth established by a motion capture system. In addition, we presented our novel high yaw rate dynamic model and it's corresponding extended Kalman Filter which proved to provide better estimates as the robot is performing dynamic maneuvers. Also, we found that the Complementary Filter and the Complementary Kalman Filter have almost identical performance under all tested situations. Remarkably, the proprietary DMP estimate from the MPU-9250 appears to very accurate over the conditions tested, even under high yaw rates, even though it is not based on the dynamic model of the physical system. The next step of this project is to implement a similar high yaw rate dynamic modeled Kalman Filter into our ball-balancing robot [3] to help us control the robot during aggressive and fast yaw maneuvers. Compared to the simpler MIP robot, the ball-balancer is highly nonlinear and the dynamics are heavily coupled under high yaw rate, which should prove to be a challenging topic.

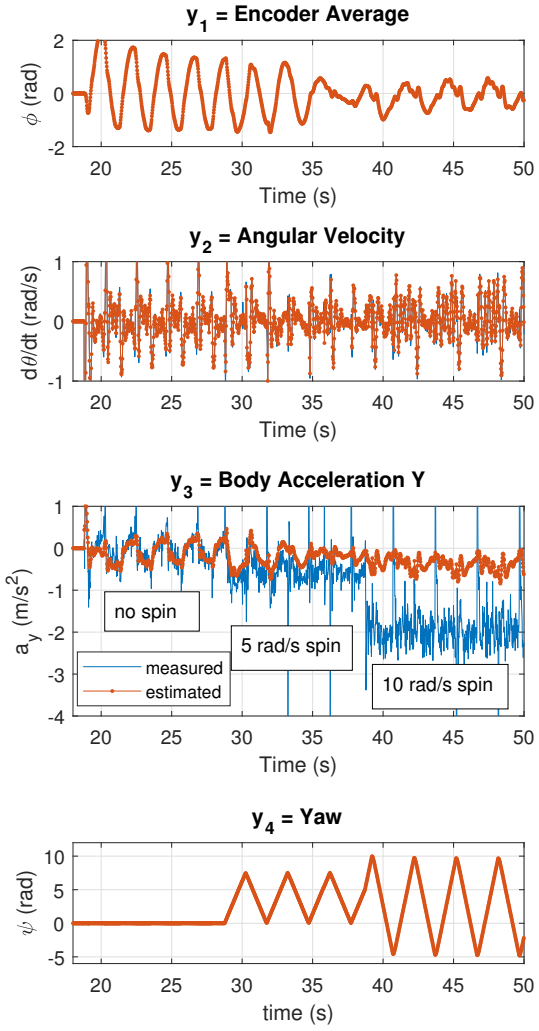

Fig. 5: Measured $\boldsymbol{y}$ vs estimated $\boldsymbol{y}$ plot for the yaw EKF during the no spin, 5 rad/s spin and 10 rad/s spin experiments. The offset in the accelerometer measurement is very likely caused by the IMU being not on the body's center of mass.

ACKNOWLEDGMENT

Special thanks to Clark Briggs at ATA Engineering for offering our team use of their motion capture system.

REFERENCES

[1] Lee, Hyungjik, and Seul Jung. "Balancing and navigation control of a mobile inverted pendulum robot using sensor fusion of low cost sensors." *Mechatronics 22.1*, pp. 95-105, (2012).

[2] Akesson, Johan, Anders Blomdell, and Rolf Braun. "Design and control of yaipAn inverted pendulum on two wheels robot." *IEEE International Conference on Control Applications*, 2006.

[3] Yang, Daniel, et al., "Design and control of a micro ball-balancing robot (MBBR) with orthogonal midlatitude omniwheel placement," in *IEEE Int'l Conf. on Intelligent Robots and Systems (IROS)*, pp. 4098-4104, 2015.

[4] Hertig, Lionel, et al. "Unified state estimation for a ballbot." *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[5] Higgins, Walter T. "A comparison of complementary and Kalman filtering." *IEEE Transactions on Aerospace and Electronic Systems*, pp. 321-325, 1975.

[6] Sabatini, Angelo M. "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing." *IEEE Transactions on Biomedical Engineering 53.7*, pp. 1346-1356, 2006.

[7] Suh, Young Soo. "Orientation estimation using a quaternion-based indirect Kalman filter with adaptive estimation of external acceleration." *IEEE Transactions on Instrumentation and Measurement 59.12*, pp. 3296-3305, 2010.

[8] Ligorio, Gabriele, and Angelo M. Sabatini. "A novel Kalman filter for human motion tracking with an inertial-based dynamic inclinometer." *IEEE Transactions on Biomedical Engineering 62.8*, pp. 2033-2043, 2015.

[9] Eide, R., Egelid, P. M., Stams, A., and Karimi, H. R. "LQG control design for balancing an inverted pendulum mobile robot." *Intelligent Control and Automation 2*, pp.160-166, 2011.

[10] Shimizu, Yuta, and Akira Shimada. "Direct tilt angle control on inverted pendulum mobile robots." *IEEE International Workshop on Advanced Motion Control*, 2010.

[11] Bewley, Thomas R. "Numerical renaissance: simulation, optimization, and control." San Diego , pp.475, 2012.

[12] Zhuo, Zhu. "LQG Controller Design of the Mobile Inverted Pendulum." Diss. University of California, San Diego, 2017.

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on December 15,2021 at 03:37:23 UTC from IEEE Xplore. Restrictions apply.